Python

The below table largely covers the TOC for 5 popular books. Learning Python (Fourth Edition) has a more in-depth look at concepts than any of the other books. However this book also does not essentially cover some aspects that are covered in other books.

No. Diving into Python The Python Standard Library by Example Python Essential Reference (4th edition) The Quick Python Book Learning Python

Introductory Concepts covering installation on different OS, version history, interpreter. This section also covers questions like Why, Who, What and Where on Python.

1 1. Installing Python
2. Which Python is right for you ?
3. Python & your OS
4. Interactive Shell
5. Summary 1. Introduction (Text) 1. Tutorial Introduction
2. Lexical Conventions and Syntax 1. About Python
2. Getting Started 1. Python Q & A Session
1. Why do people use Python ?
2. Downside of using it
3. Who uses Python Today ?
4. What Can I do with Python ?
5. Python vs Language X
6. Test your Knowledge
2. How Python runs programs
1. Python Interpreter
2. Program Execution
1. Programmer View
2. Python View
3. Execution Model Variations
1. Implementation Alternatives
2. Execution Optimization Tools
3. Frozen Binaries
3. How you run programs
1. Interactive prompt
2. Your first script
<snipped>

Python Object Types, Numeric Types, Data Structures, Control Structures, Scopes and Arguments

2 1. Your first program
2. Declaring Functions
3. Python Data types vs Other Languages
4. Documenting Functions
5. Everything is an Object
6. The Import Search Path
7. What is an Object ?
8. Indenting Code
9. Testing Modules
10. Native Datatypes
1. Dictionaries
2. List
3. Tuples
11. Variables & referencing 1. Data Structures 1. Types and Objects
2. Operators and Expressions
3. Program Structure and Control Flow
4. Functions and Functional Programming

5. Classes and Object Oriented Programming
6. Modules, Packages and Distribution
7. Input and Output
8. Execution Environment
9. Testing, Debugging, Profiling and Tuning
<Covered further along in the book>
Data Structures, Algorithms & Code simplification
String & Text Handling 1. Python Overview
1. Built-in Data types
2. Control Structures
3. Module
4. OOPs
2. Basics
1. Lists
2. Dictionaries
3. Tuple
4. Sets
5. Strings
6. Control Flow
3. Functions
4. Modules and Scoping Rules
5. Python Programs 1. Introducing Python Object Types
1. Why use built-in Types ?
2. Core data types
3. Numbers, Lists, Dictionaries, Tuples, Files, Other Core Types
4. User Defined Classes
2. Numeric Types
1. Literals, Built-in tools, expression operators
2. Formats, Comparisons, Division, Precision
3. Complex Numbers
4. Hexadecimal, Octal & Binary
5. Bitwise Operations
6. Decimal, Fraction, Sets, Booleans
<Covered further along in the book>
1. Statements & Syntax
2. Assignments, Expressions & Syntax
3. If Tests & Syntax Rules
4. Scopes
5. Arguments
Built-in functions, Function Design, Recursive Functions, Introspection, Annotations, Lambda, Filter and Reduce
3 1. Power of Introspection
1. Optional and Named Arguments
2. type, str, dir and other built-in functions
3. Object References with getattr
4. Filtering Lists
5. Lambda Functions
6. Real world Lambda functions
None 1. Built-in functions
2. Python run-time services None Built-in functions are covered as part of the topic above but from a numeric perspective
1. Advanced Function Topics
1. Function Design

2. Recursive Functions
3. Attributes and Annotation
4. Lambda
5. Mapping Functions over sequences
6. Filter and Reduce
<Covered further along in the book>
Special Class Attributes
Display Tool
OOPS, Modules
4 1. Objects and Object Orientation
1. Importing Modules
2. Defining Classes
3. Initializing and Coding Classes
4. Self & __init__
5. Instantiating Classes
6. Garbage Collection
7. Wrapper Classes
8. Special Class Methods
9. Advanced Class Methods
10. Class Attributes
11. Private Functions None Covered partially section 2 1. Packages
2. Data Types and Objects
3. Advanced Object Oriented Features 1. Modules
1. Why use Modules ?
2. Program Architecture
3. Module Search Path
4. Module Creation & Usage
5. Namespaces
6. Reloading Modules
7. Packages
2. Advanced Module Topics
1. Data Hiding in Modules
2. as Extension for import and from
3. Modules are Objects: Metaprograms
4. Transitive Module Reloads
5. Module Design Concepts
6. Module Gotchas
3. OOP
1. Why use classes ?
2. Classes & Instances
3. Attribute Inheritance Search
4. Class Method Calls
5. Class Trees
6. Class Objects & Default Behavior
7. Instance Objects are Concrete Items
8. Intercepting Python Operators
9. Classes Vs. Dictionaries
10. Class customization by Inheritance
11. Operator Overloading
12. Subclasses
13. Polymorphism in Action
14. Designing with Classes

15. Mix-in Classes
Advanced Class Topics
5 None None None None 1. Advanced Class Topics
1. Extending Types by Embedding
2. Extending Types by Subclassing
3. Static and Class Methods
4. Decorators and Metaclasses
5. Class Gotchas
Exceptions
6 1. Exceptions and File Handling
1. Handling Exceptions
2. Using exceptions for other purposes 1. Exceptions 1. Exceptions Basics
1. Why use Exceptions ?
2. Default Exception Handler
3. User-Defined Exceptions
4. Class Based Exceptions
5. Designing with Exceptions
XML, HTTP, SOAP, Network Programming, I18N, Unicode
7 1. Regular Expressions
2. Parsing / Processing Mark-up languages (HTML, XML)
1. Unicode
3. HTTP Web Services
1. Headers
2. Debugging
4. SOAP Web Services 1. Networking
2. Internet
3. Email
4. Internationalization and Localization 1. Network Programming and Sockets
2. Internet Application Programming
3. Web Programming
4. Internet Data Handling & Encoding 1. Network, web programming 1. Unicode and Bytes Strings
Miscellaneous
8 None 1. Algorithms
2. Cryptography
3. Data compression and archiving
4. Processes and Threads
5. Data persistence & exchange 1. Extending & Embedding Python 1. GUI None